



Fachbereich Computertechnik

IBM-PS/2

Tastatur Treiber

Benutzeranleitung

Pascal Näf, Fabian Heusser

Dozent: Alex Klapproth

History

Nr	Datum	Name	Beschreibung
1	2002-06-20	hef	Initial Version
2	2002-06-24	nap	Korrekturen

Status

approved final

Inhalt

1	Distribution	4
2	Voraussetzungen	4
3	Anschluss.....	5
3.1	Anschluss mit Adapter.....	5
3.2	Anschluss ohne Adapter	5
3.3	Steckerbelegung Tastaturstecker	6
4	Importieren des Modules	6
5	Funktionen	7
5.1	Vom Keyboard lesen	7
5.2	Zum Keyboard schreiben	7
5.3	Test- und Initialisierungsfunktionen	7
6	Testmodul „ps2demo.c“	7
6.1	Voraussetzungen.....	7
6.2	Funktion	7

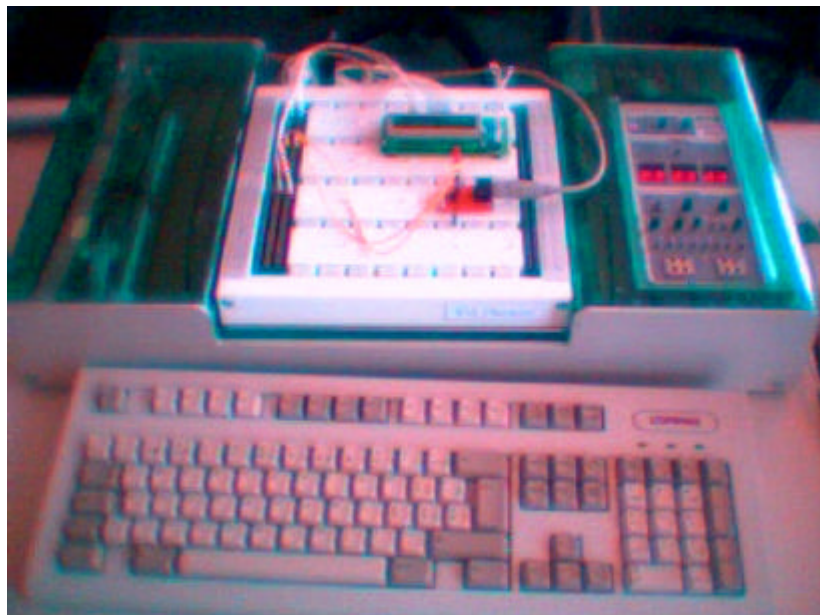


Abbildung 1 – Übersicht Anordnung

1 Distribution

Unsere PS/2 Tastatur-Treiber Distribution besteht aus den folgenden Dateien:

- ps2drv.c - Den Sourcecode des Treibers in C.
- ps2drv.h - Das Treiber-Modul zum Importieren.
- ps2demo.c - Demo/Test Modul, welches ein Test Programm enthält.
- ps2.zip - Metrowerks-Projekt mit LCD- und Tastatur-Treiber, Demomodul und Benutzeranleitung.
- userguide.pdf - Dieses Dokument.

2 Voraussetzungen

Diese Anleitung setzt die HTA-Luzern Mikrocontroller Entwicklungsumgebung, bestehend aus Medusa-Trainer und dem Metrowerks CodeWarrior IDE, voraus. Weiter sind im Minimum die Files ,ps2drv.c" und ,ps2drv.h' nötig.

Die Testanwendung „ps2demo“ benötigt zusätzlich das LCD-Display mit den entsprechenden Treibern.

Alle diese Dateien sind im ps2.zip zu einem Metrowerks Projekt zusammengefasst, das einen einfachen Einsatz ermöglicht.

Der Adapter zum Anschliessen bringt Komfort, kann aber leicht selbst auf dem Entwicklungsbord realisiert werden (siehe Abschnitt 3).

3 Anschluss

3.1 Anschluss mit Adapter

Zum bequemen Anschliessen einer Tastatur haben wir einen Adapter gebaut. Dieser wird mit vier Kabeln wie folgt am Medusa Trainer angeschlossen:

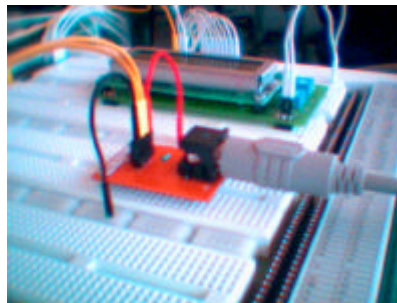
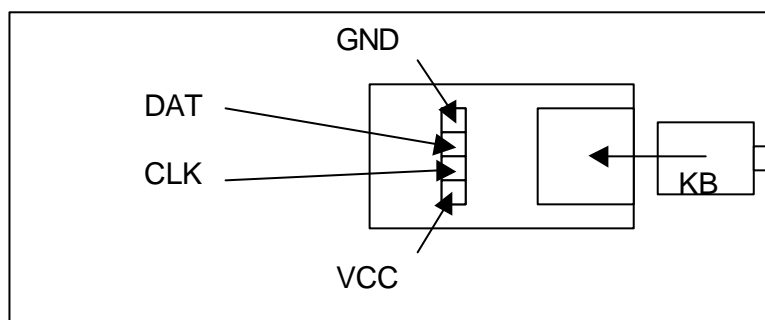


Abbildung 2 - Anschluss



Der Treiber ist so eingestellt, dass man DAT und CLK wie folgt an den Medusa-Trainer anschliessen muss:

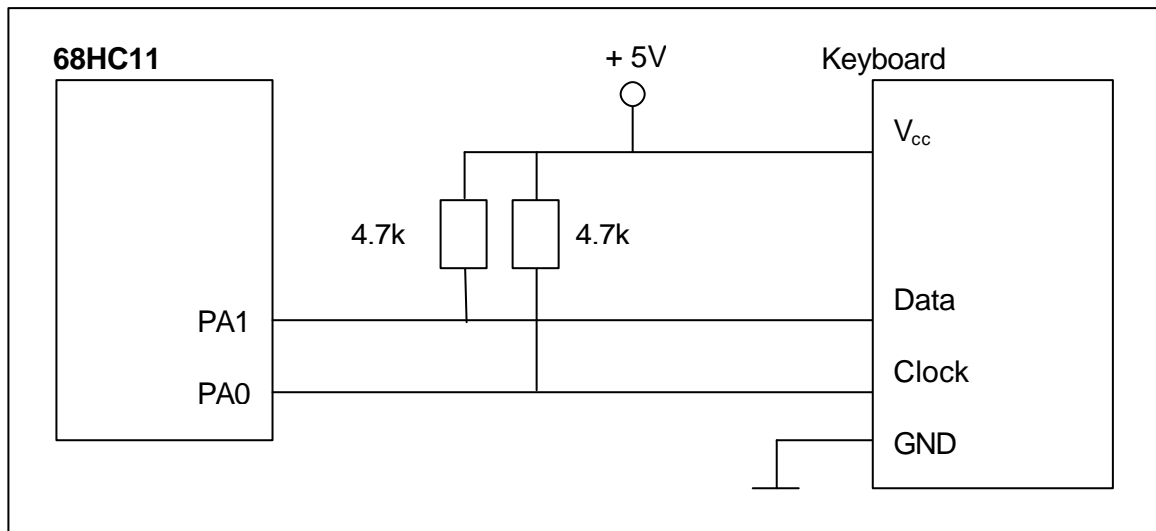
CLK	PA0
DAT	PA1

Der Programmierer kann diese Belegung einfach ändern. Hier die betreffende Stelle aus dem oberen Teil des `,ps2drv.c'`:

```
#define KeyBoard  PORTA_  
#define Clock    PA0  
#define Data     PA1  
  
#define Dir      DDRA_  
#define ClockDir DDA0  
#define DataDir  DDA1
```

3.2 Anschluss ohne Adapter

Soll eine Tastatur ohne den Adapter angeschlossen werden, kann dies mit einem geringen Mehraufwand bewerkstelligt werden. Das nachfolgende Diagramm zeigt das Schema der normalen Belegung:

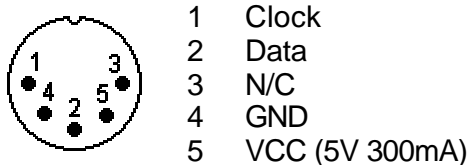


Das obige Schema entspricht dem Adapter, ausser dass bei diesem der Stecker und die Widerstände fest verdrahtet sind. Auch hat der Adapter Buchsen zum Anschluss der Kabel.

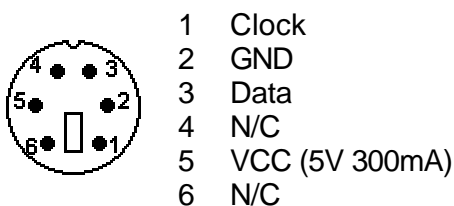
3.3 Steckerbelegung Tastaturstecker

Die Tastaturstecker sind wie folgt belegt:

5 Pol DIN Stecker (AT)



6 Pol Mini-DIN Stecker (PS/2)



Achtung: Die obigen Pin-Nummerierung trifft zu, wenn man von vorne auf den Stecker schaut. Lötet man eine Buchse an, ist es spiegelverkehrt (von vorne in die Buchse gesehen).

4 Importieren des Modules

Die folgenden Anweisung importiert die Funktionen des Tastatur-Treibers, vorausgesetzt die Files `ps2drv.c` und `ps2drv.h` befinden sich im selben Verzeichnis wie die C Datei. Also im `sources/` Verzeichnis eines Metrowerks Projekt.

```
#include "ps2drv.h"
```

5 Funktionen

Nachfolgend eine kurze Zusammenstellung der verfügbaren Funktionen. Genauere Spezifikationen der Funktionsweise befindet sich im Dokument „Software Requirement Specification“ und in den Source-Files „ps2drv.h“ und „ps2drv.c“.

5.1 Vom Keyboard lesen

`unsigned char getch ();` - Liefert den ASCII-Code der gedrückten Taste.
`unsigned char getScanCode();` - Liest ein Byte von der Tastatur (+ Errorhandling).
`unsigned char read();` - Low-Level Funktion; liest ein Byte von der Tastatur.

5.2 Zum Keyboard schreiben

`void setLED(vartchar code);` - Setzt die LEDs auf der Tastatur.
`void write(unsigned char);` - Low-Level Funktion; Sendet ein Byte zur Tastatur.

5.3 Test- und Initialisierungsfunktionen

`unsigned char testKB();` - Testet das Keyboard.
`void reset();` - Setzt das Keyboard zurück.
`void init();` - Initialisiert den Treiber.
`unsigned char extededInit();` - Initialisiert den Treiber, setzt das Keyboard zurück und testet es.

6 Testmodul „ps2demo.c“

Um die Funktionsweise des Treibers zu testen und die Funktionen zu demonstrieren, entstand das Modul ‚ps2demo.c‘. Es demonstriert die grundlegenden Funktionen des Tastatortreibers und erleichtert den Einstieg erheblich. Es steht auch ein gezipptes Metrowerks Projekt für unsere HTA - Entwicklungsumgebung zur Verfügung, mit welchem das Demo schnell funktioniert.

6.1 Voraussetzungen

Zusätzlich zu den unter Abschnitt 2 erwähnten Voraussetzungen setzt das PS2 Demo den Medusa-Trainer mit aufgestecktem und korrekt angeschlossenem LCD-Display voraus.

6.2 Funktion

Wird das Programm gestartet erscheint der Keyboardtest auf dem Display. Dieser endet mit OK, wenn das Programm die Tastatur erfolgreich initialisierte. Die Tastatur sollte dabei den Reset signalisieren, indem ihre 3 LEDs kurz aufleuchten. Danach stehen folgende Funktionen zur Verfügung.

- mit F12: Initialisierung mit Test wie beim Starten.
- Tasten a-z und 0-9 erscheinen auf dem Display.
- Backspace und Return funktionieren erwartungsgemäss.
- Capslock, Numlock und Scrolllock schalten die entsprechenden LEDs ein oder aus.